Joint Space Neural Probabilistic Language Model for Statistical Machine Translation

Anonymous Author(s)
Affiliation
Address
email

Abstract

A neural probabilistic language model (NPLM) provides an idea to achieve the better perplexity than n-gram language model and their smoothed language models. This paper investigates application area in bilingual NLP, specifically Statistical Machine Translation (SMT). We focus on the perspectives that NPLM has potential to open the possibility to complement potentially 'huge' monolingual resources into the 'resource-constraint' bilingual resources. We introduce an ngram-HMM language model as NPLM using the non-parametric Bayesian construction. In order to facilitate the application to various tasks, we propose the joint space model of ngram-HMM language model. We show an experiment of system combination in the area of SMT. One discovery was that our treatment of noise improved the results 0.20 BLEU points if NPLM is trained in relatively small corpus, in our case 500,000 sentence pairs, which is often the case due to the long training time of NPLM.

1 Introduction

A neural probabilistic language model (NPLM) [3, 4] and the distributed representations [25] provide an idea to achieve the better perplexity than n-gram language model [47] and their smoothed language models [26, 9, 48]. Recently, the latter one, i.e. smoothed language model, has had a lot of developments in the line of nonparametric Bayesian methods such as hierarchical Pitman-Yor language model (HPYLM) [48] and Sequence Memoizer (SM) [51, 20], including an application to SMT [36, 37, 38]. A NPLM considers the representation of data in order to make the probability distribution of word sequences more compact where we focus on the similar semantical and syntactical roles of words. For example, when we have two sentences *"The cat is walking in the bedroom"* and *"A dog was running in a room"*, these sentences can be more compactly stored than the n-gram language model if we focus on the similarity between (the, a), (bedroom, room), (is, was), and (running, walking). Thus, a NPLM provides the semantical and syntactical roles of words as a language model. A NPLM of [3] implemented this using the multi-layer neural network and yielded 20% to 35% better perplexity than the language model with the modified Kneser-Ney methods [9].

There are several successful applications of NPLM [41, 11, 42, 10, 12, 14, 43]. First, one category of applications include POS tagging, NER tagging, and parsing [12, 7]. This category uses the features provided by a NPLM in the limited window size. It is often the case that there is no such long range effects that the decision cannot be made beyond the limited windows which requires to look carefully the elements in a long distance. Second, the other category of applications include Semantic Role Labeling (SRL) task [12, 14]. This category uses the features within a sentence. A typical element is the predicate in a SRL task which requires the information which sometimes in a long distance but within a sentence. Both of these approaches do not require to obtain the best tag sequence, but these tags are independent. Third, the final category includes MERT process [42] and possibly many others where most of them remain undeveloped. The objective of this learning

in this category is not to search the best tag for a word but the best sequence for a sentence. Hence, we need to apply the sequential learning approach. Although most of the applications described in [11, 10, 12, 14] are monolingual tasks, the application of this approach to a bilingual task introduces really astonishing aspects, which we can call “creative words” [50], automatically into the traditional resource constrained SMT components. For example, the training corpus of word aligner is often strictly restricted to the given parallel corpus. However, a NPLM allows this training with huge monolingual corpus. Although most of this line has not been even tested mostly due to the problem of computational complexity of training NPLM, [43] applied this to MERT process which reranks the n-best lists using NPLM. This paper aims at different task, a task of system combination [1, 29, 49, 15, 13, 35]. This category of tasks employs the sequential method such as Maximum A Posteriori (MAP) inference (Viterbi decoding) [27, 44, 33] on Conditional Random Fields (CRFs) / Markov Random Fields (MRFs).

Although this paper discusses an ngram-HMM language model which we introduce as one model of NPLM where we borrow many of the mechanism from infinite HMM [19] and hierarchical Pitman-Yor LM [48], one main contribution would be to show one new application area of NPLM in SMT. Although several applications of NPLM have been presented, there have been no application to the task of system combination as far as we know.

The remainder of this paper is organized as follows. Section 2 describes ngram-HMM language model while Section 3 introduces a joint space model of ngram-HMM language model. In Section 4, our intrinsic experimental results are presented, while in Section 5 our extrinsic experimental results are presented. We conclude in Section 5.

2 Ngram-HMM Language Model

Generative model Figure 1 depicted an example of ngram-HMM language model, i.e. 4-gram-HMM language model in this case, in blue (in the center). We consider a Hidden Markov Model (HMM) [40, 21, 2] of size K which emits n-gram word sequence $w_i, \dots, w_{i-K+1}$ where $h_i, \dots, h_{i-K+1}$ denote corresponding hidden states. The arcs from w_{i-3} to w_i, $\cdots$, w_{i-1} to w_i show the back-off relations appeared in language model smoothing, such as Kneser-Ney smoothing [26], Good-Turing smoothing [24], and hierarchical Pitman-Yor LM smoothing [48].

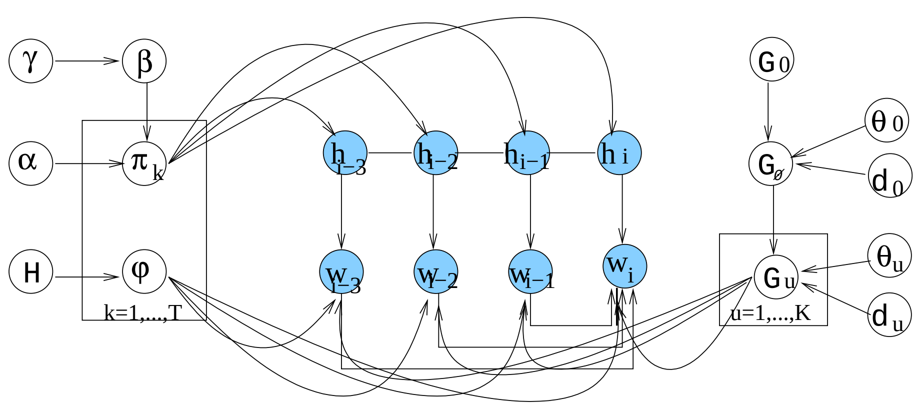

Figure 1: Figure shows a graphical representation of the 4-gram HMM language model.

In the left side in Figure 1, we place one Dirichlet Process prior DP(α, H), with concentration parameter α and base measure H, for the transition probabilities going out from each hidden state. This construction is borrowed from the infinite HMM [2, 19]. The observation likelihood for the hidden word h_t are parameterized as in $w_t | h_t \sim F(\phi_{h_t})$ since the hidden variables of HMM is limited in its representation power where ϕ_{h_t} denotes output parameters. This is since the observations can be regarded as being generated from a dynamic mixture model [19] as in (1), the Dirichlet priors

on the rows have a shared parameter.

$$p(w_i|h_{i-1}=k) = \sum_{h_i=1}^{K} p(h_i|h_{i-1}=k)p(w_i|h_i)$$
$$= \sum_{h_i=1}^{K} \pi_{k,h_i} p(w_i|\phi_{h_i}) \quad (1)$$

In the right side in Figure 1, we place Pitman-Yor prior PY, which has advantage in its power-law behavior as our target is NLP, as in (2):

$$w_i|w_{1:i-1} \sim \mathrm{PY}(d_i, \theta_i, G_i) \quad (2)$$

where α is a concentration parameter, θ is a strength parameter, and G_i is a base measure. This construction is borrowed from hierarchical Pitman-Yor language model [48].

Inference We compute the expected value of the posterior distribution of the hidden variables with a beam search [19]. This blocked Gibbs sampler alternate samples the parameters (transition matrix, output parameters), the state sequence, hyper-parameters, and the parameters related to language model smoothing. As is mentioned in [19], this sampler has characteristic in that it adaptively truncates the state space and run dynamic programming as in (3):

$$p(h_t|w_{1:t}, u_{1:t}) = p(w_t|h_t) \sum_{h_{t-1}:u_t<\pi^{(h_{t-1},h_t)}} p(h_{t-1}|w_{1:t-1}, u_{1:t-1}) \quad (3)$$

where u_t is only valid if this is smaller than the transition probabilities of the hidden word sequence $h_1, \ldots, h_K$. Note that we use an auxiliary variable u_i which samples for each word in the sequence from the distribution $u_i \sim \mathrm{Uniform}(0, \pi^{(h_{i-1},h_i)})$. The implementation of the beam sampler consists of preprocessing the transition matrix π and sorting its elements in descending order.

Initialization First, we obtain the parameters for hierarchical Pitman-Yor process-based language model [48, 23], which can be obtained using a block Gibbs sampling [32].

Second, in order to obtain a better initialization value h for the above inference, we perform the following EM algorithm instead of giving the distribution of h randomly. This EM algorithm incorporates the above mentioned truncation [19]. In the E-step, we compute the expected value of the posterior distribution of the hidden variables. For every position h_i, we send a forward message $\alpha(h_{i-n+1:i-1})$ in a single path from the start to the end of the chain (which is the standard forward recursion in HMM; Hence we use α). Here we normalize the sum of α considering the truncated variables $u_{i-n+1:i-1}$.

$$\alpha(h_{i-n+2:i}) = \frac{\sum \alpha(h_{i-n+1:i-1})}{\sum \alpha(u_{i-n+1:i-1})} P(w_i|h_i) \sum \alpha(u_{i-n+1:i-1}) P(h_i|h_{i-n+1:i-1}) \quad (4)$$

Then, for every position h_j, we send a message $\beta(h_{i-n+2:i}, h_j)$ in multiple paths from the start to the end of the chain as in (5),

$$\beta(h_{i-n+2:i}, h_j) = \frac{\sum \alpha(h_{i-n+1:i-1})}{\sum \alpha(u_{i-n+1:i-1})} P(w_i|h_i) \sum \beta(h_{i-n+1:i-1}, h_j) P(h_i|h_{i-n+1:i-1}) \quad (5)$$

This step aims at obtaining the expected value of the posterior distribution (Similar construction to use expectation can be seen in factored HMM [22]). In the M-step, using this expected value of the posterior distribution obtained in the E-step to evaluate the expectation of the logarithm of the complete-data likelihood.

3 Joint Space Model

In this paper, we mechanically introduce a joint space model. Other than the ngram-HMM language model obtained in the previous section, we will often encounter the situation where we have another hidden variables h^1 which is irrelevant to h^0 which is depicted in Figure 2. Suppose that we have

the ngram-HMM language model yielded the hidden variables suggesting semantic and syntactical role of words. Adding to this, we may have another hidden variables suggesting, say, a genre ID. This genre ID can be considered as the second context which is often not closely related to the first context. This also has an advantage in this mechanical construction that the resulted language model often has the perplexity smaller than the original ngram-HMM language model. Note that we do not intend to learn this model jointly using the universal criteria, but we just concatenate the labels by different tasks on the same sequence. By this formulation, we intend to facilitate the use of this language model.

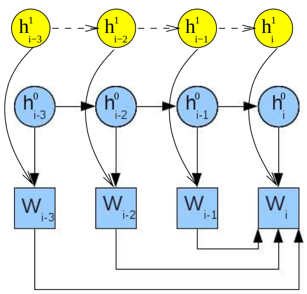

Figure 2: Figure shows the joint space 4-gram HMM language model.

It is noted that those two contexts may not be derived in a single learning algorithm. For example, language model with the sentence context may be derived in the same way with that with the word context. In the above example, a hidden semantics over sentence is not a sequential object. Hence, this can be only considering all the sentence are independent. Then, we can obtain this using, say, LDA.

4 Intrinsic Evaluation

We compared the perplexity of ngram-HMM LM (1 feature), ngram-HMM LM (2 features, the same as in this paper and genre ID is 4 class), modified Kneser-Ney smoothing (irstlm) [18], and hierarchical Pitman Yor LM [48]. We used news2011 English testset. We trained LM using Europarl.

	ngram-HMM (1 feat)	ngram-HMM (2 feat)	modified Kneser-Ney	hierarchical PY
Europarl 1500k	114.014	113.450	118.890	118.884

Table 1: Table shows the perplexity of each language model.

5 Extrinsic Evaluation: Task of System Combination

We applied ngram-HMM language model to the task of system combination. For given multiple Machine Translation (MT) outputs, this task essentially combines the best fragments among given MT outputs to recreate a new MT output. The standard procedure consists of three steps: Minimum Bayes Risk decoding, monolingual word alignment, and monotonic consensus decoding. Although these procedures themselves will need explanations in order to understand the following, we keep the main text in minimum, moving some explanations (but not sufficient) in appendices. Note that although this experiment was done using the ngram-HMM language model, any NPLM may be sufficient for this purpose. In this sense, we use the term NPLM instead of ngram-HMM language model.

Features in Joint Space The first feature of NPLM is the semantically and syntactically similar words of roles, which can be derived from the original NPLM. We introduce the second feature in this paragraph, which is a genre ID.

The motivation to use this feature comes from the study of domain adaptation for SMT where it becomes popular to consider the effect of genre in testset. This paper uses Latent Dirichlet Allocation

(LDA) [5, 46, 6, 45, 33] to obtain the genre ID via (unsupervised) document classification since our interest here is on the genre of sentences in testset. And then, we place these labels on a joint space.

LDA represents topics as multinomial distributions over the W unique word-types in the corpus and represents documents as a mixture of topics. Let C be the number of unique labels in the corpus. Each label c is represented by a W-dimensional multinomial distribution ϕ_c over the vocabulary. For document d, we observe both the words in the document $w^{(d)}$ as well as the document labels $c^{(d)}$. Given the distribution over topics θ_d, the generation of words in the document is captured by the following generative model.The parameters α and β relate to the corpus level, the variables θ_d belong to the document level, and finally the variables z_{dn} and w_{dn} correspond to the word level, which are sampled once for each word in each document.

Using topic modeling in the second step, we propose the overall algorithm to obtain genre IDs for testset as in (5).

1. Fix the number of clusters C, we explore values from small to big where the optimal value will be searched on tuning set.
2. Do unsupervised document classification (or LDA) on the source side of the tuning and test sets.
   (a) For each label $c \in \{1, \dots C\}$, sample a distribution over word-types $\phi_c \sim$ **Dirichlet**$(\cdot|\beta)$
   (b) For each document $d \in \{1, \dots, D\}$
      i. Sample a distribution over its observed labels $\theta_d \sim$ **Dirichlet**$(\cdot|\alpha)$
      ii. For each word $i \in \{1, \dots, N_d^W\}$
         A. Sample a label $z_i^{(d)} \sim$ **Multinomial**(θ_d)
         B. Sample a word $w_i^{(d)} \sim$ **Multinomial**(ϕ_c) from the label $c = z_i^{(d)}$
3. Separate each class of tuning and test sets (keep the original index and new index in the allocated separated dataset).
4. (Run system combination on each class.)
5. (Reconstruct the system combined results of each class preserving the original index.)

Modified Process in System Combination Given a joint space of NPLM, we need to specify in which process of the task of system combination among three processes use this NPLM. We only discuss here the standard system combination using confusion-network. This strategy takes the following three steps (Very brief explanation of these three is available in Appendix):

- Minimum Bayes Risk decoding [28] (with Minimum Error Rate Training (MERT) process [34])

$$\hat{E}_{best}^{MBR} = \text{argmin}_{E' \in \mathcal{E}} R(E') = \text{argmin}_{E' \in \mathcal{E}} \sum_{E' \in \mathcal{E}_E} L(E, E') P(E|F)$$
$$= \text{argmin}_{E' \in \mathcal{E}} \sum_{E' \in \mathcal{E}_E} (1 - BLEU_E(E')) P(E|F)$$

- Monolingual word alignment
- (Monotone) consensus decoding (with MERT process)

$$E_{best} = \arg\max_e \prod_{i=1}^{I} \phi(i|\bar{e}_i) p_{LM}(e)$$

Similar to the task of n-best reranking in MERT process [43], we consider the reranking of nbest lists in the third step of above, i.e. (monotone) consensus decoding (with MERT process). We do not discuss the other two processes in this paper.

On one hand, we intend to use the first feature of NPLM, i.e. the semantically and syntactically similar role of words, for paraphrases. The n-best reranking in MERT process [43] alternate the

probability suggested by word sense disambiguation task using the feature of NPLM, while we intend to add a sentence which replaces the words using NPLM. On the other hand, we intend to use the second feature of NPLM, i.e. the genre ID, to split a single system combination system into multiple system combination systems based on the genre ID clusters. In this perspective, the role of these two feature can be seen as independent. We conducted four kinds of settings below.

(A) —First Feature: N-Best Reranking in Monotonic Consensus Decoding without Noise – NPLM plain In the first setting for the experiments, we used the first feature without considering noise. The original aim of NPLM is to capture the semantically and syntactically similar words in a way that a latent word depends on the context. We will be able to get variety of words if we condition on the fixed context, which would form paraphrases in theory.

We introduce our algorithm via a word sense disambiguation (WSD) task which selects the right disambiguated sense for the word in question. This task is necessary due to the fact that a text is natively ambiguous accommodating with several different meanings. The task of WSD [14] can be written as in (6):

$$P(\text{synset}_i|\text{features}_i,\theta) \quad = \quad \frac{1}{Z(\text{features})}\prod_m g(\text{synset}_i,k)^{f(\text{feature}_i^k)} \tag{6}$$

where k ranges over all possible features, $f(\text{feature}_i^k)$ is an indicator function whose value is 1 if the feature exists, and 0 otherwise, $g(\text{synset}_i,k)$ is a parameter for a given synset and feature, θ is a collection of all these parameters in $g(\text{synset}_i,k)$, and Z is a normalization constant. Note that we use the term "synset" as an analogy of the WordNet [30]: this is equivalent to "sense" or "meaning". Note also that NPLM will be included as one of the features in this equation. If features include sufficient statistics, a task of WSD will succeed. Otherwise, it will fail. We do reranking of the outcome of this WSD task.

On the one hand, the paraphrases obtained in this way have attractive aspects that can be called "a creative word" [50]. This is since the traditional resource that can be used when building a translation model by SMT are constrained on parallel corpus. However, NPLM can be trained on huge monolingual corpus. On the other hand, unfortunately in practice, the notorious training time of NPLM only allows us to use fairly small monolingual corpus although many papers made an effort to reduce it [31]. Due to this, we cannot ignore the fact that NPLM trained not on a huge corpus may be affected by noise. Conversely, we have no guarantee that such noise will be reduced if we train NPLM on a huge corpus. It is quite likely that NPLM has a lot of noise for small corpora. Hence, this paper also needs to provide the way to overcome difficulties of noisy data. In order to avoid this difficulty, we limit the paraphrase only when it includes itself in high probability.

(B)— First Feature: N-Best Reranking in Monotonic Consensus Decoding with Noise – NPLM dep In the second setting for our experiment, we used the first feature considering noise. Although we modified a suggested paraphrase without any intervention in the above algorithm, it is also possible to examine whether such suggestion should be adopted or not. If we add paraphrases and the resulted sentence has a higher score in terms of the modified dependency score [39] (See Figure 3), this means that the addition of paraphrases is a good choice. If the resulted score decreases, we do not need to add them. One difficulty in this approach is that we do not have a reference which allows us to score it in the usual manner. For this reason, we adopt the *naive way* to deploy the above and we deploy this with *pseudo references*. (This formulation is equivalent that we decode these inputs by MBR decoding.) First, if we add paraphrases and the resulted sentence does not have a very bad score, we add these paraphrases since these paraphrase are not very bad (*naive* way). Second, we do scoring between the sentence in question with *all the other candidates* (*pseudo references*) and calculate an average of them. Thus, our second algorithm is to select a paraphrase which may not achieve a very bad score in terms of the modified dependency score using NPLM.

(C) — Second Feature: Genre ID — DA (Domain Adaptation) In the third setting of our experiment, we used only the second feature. As is mentioned in the explanation about this feature, we intend to splits a single module of system combination into multiple modules of system combi-

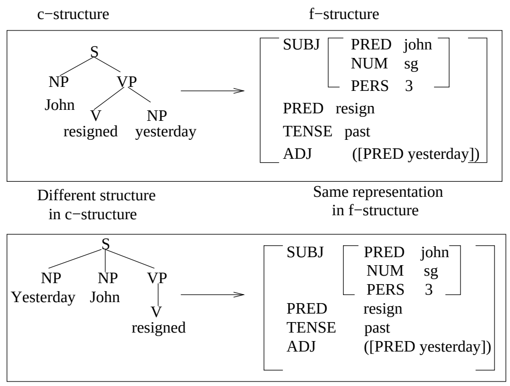

Figure 3: By the modified dependency score [39], the score of these two sentences, "John resigned yesterday" and "Yesterday John resigned", are the same. Figure shows c-structure and f-structure of two sentences using Lexical Functional Grammar (LFG) [8].

nation according to the genre ID. Hence, we will use the module of system combination tuned for the specific genre ID, [1].

(D) — First and Second Feature — COMBINED In the fourth setting we used both features. In this setting, (1) we used modules of system combination which are tuned for the specific genre ID, and (2) we prepared NPLM whose context can be switched based on the specific genre of the sentence in test set. The latter was straightforward since these two features are stored in joint space in our case.

Experimental Results ML4HMT-2012 provides four translation outputs (*s1* to *s4*) which are MT outputs by two RBMT systems, APERTIUM and LUCY, PB-SMT (MOSES) and HPB-SMT (MOSES), respectively. The tuning data consists of 20,000 sentence pairs, while the test data consists of 3,003 sentence pairs.

Our experimental setting is as follows. We use our system combination module [16, 17, 35], which has its own language modeling tool, MERT process, and MBR decoding. We use the BLEU metric as loss function in MBR decoding. We use TERP[2] as alignment metrics in monolingual word alignment. We trained NPLM using 500,000 sentence pairs from English side of EN-ES corpus of EUROPARL[3].

The results show that the first setting of NPLM-based paraphrased augmentation, that is NPLM plain, achieved 25.61 BLEU points, which lost 0.39 BLEU points absolute over the standard system combination. The second setting, NPLM dep, achieved slightly better results of 25.81 BLEU points, which lost 0.19 BLEU points absolute over the standard system combination. Note that the baseline achieved 26.00 BLEU points, the best single system in terms of BLEU was s4 which achieved 25.31 BLEU points, and the best single system in terms of METEOR was s2 which achieved 0.5853. The third setting achieved 26.33 BLEU points, which was the best among our four settings. The fourth setting achieved 25.95, which is again lost 0.05 BLEU points over the standard system combination.

Other than our four settings where these settings differ which features to use, we run several different settings of system combination in order to understand the performance of four settings. Standard system combination using BLEU loss function (line 5 in Table 2), standard system combination using TER loss function (line 6), system combination whose backbone is unanamously taken from the RBMT outputs (MT input s2 in this case; line 11), and system combination whose backbone is selected by the modified dependency score (which has three variations in the figure; modDep preci-

[1] E.g., we translate newswire with system combination module tuned with newswire tuning set, while we translate medical text with system combination module tuned with medical text tuning set.

[2] *http://www.cs.umd.edu/∼snover/terp*

[3] *http://www.statmt.org/europarl*

sion, recall and Fscore; line 12, 13 and 14). One interesting characteristics is that the s2 backbone (line 11) achieved the best score among all of these variations. Then, the score of the modified dependency measure-selected backbone follows. From these runs, we cannot say that the runs related to NPLM, i.e. (A), (B) and (D), were not particularly successful. The possible reason for this was that our interface with NPLM was only limited to paraphrases, which was not very successfuly chosen by reranking.

	NIST	BLEU	METEOR	WER	PER
MT input s1	6.4996	0.2248	0.5458641	64.2452	49.9806
MT input s2	6.9281	0.2500	0.5853446	62.9194	48.0065
MT input s3	7.4022	0.2446	0.5544660	58.0752	44.0221
MT input s4	7.2100	0.2531	0.5596933	59.3930	44.5230
standard system combination (BLEU)	7.6846	0.2600	0.5643944	56.2368	41.5399
standard system combination (TER)	7.6231	0.2638	0.5652795	56.3967	41.6092
(A) NPLM plain	7.6041	0.2561	0.5593901	56.4620	41.8076
(B) NPLM dep	7.6213	0.2581	0.5601121	56.1334	41.7820
(C) DA	7.7146	0.2633	0.5647685	55.8612	41.7264
(D) COMBINED	7.6464	0.2595	0.5610121	56.0101	41.7702
s2 backbone	7.6371	0.2648	0.5606801	56.0077	42.0075
modDep precision	7.6670	0.2636	0.5659757	56.4393	41.4986
modDep recall	7.6695	0.2642	0.5664320	56.5059	41.5013
modDep Fscore	7.6695	0.2642	0.5664320	56.5059	41.5013

Table 2: This table shows single best performance, the performance of the standard system combination (BLEU and TER loss functions), the performance of four settings in this paper ((A),. . .,(D)), the performance of s2 backboned system combination, and the performance of the selection of sentences by modified dependency score (precision, recall, and F-score each).

Conclusion and Perspectives

This paper proposes a non-parametric Bayesian way to interpret NPLM, which we call ngram-HMM language model. Then, we add a small extension to this by concatenating other context in the same model, which we call a joint space ngram-HMM language model. The main issues investigated in this paper was an application of NPLM in bilingual NLP, specifically Statistical Machine Translation (SMT). We focused on the perspectives that NPLM has potential to open the possibility to complement potentially 'huge' monolingual resources into the 'resource-constraint' bilingual resources. We compared our proposed algorithms and others. One discovery was that when we use a fairly small NPLM, noise reduction may be one way to improve the quality. In our case, the noise reduced version obtained 0.2 BLEU points better.

Further work would be to apply this NPLM in various other tasks in SMT: word alignment, hierarchical phrase-based decoding, and semantic incorporated MT systems in order to discover the merit of 'depth' of architecture in Machine Learning.

References

[1] Bangalore, S., Bordel, G., and Riccardi, G. Computing consensus translation from multiple machine translation systems. *In Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)* (2001), 350–354.

[2] Beal, M. J. Variational algorithms for approximate bayesian inference. *PhD Thesis at Gatsby Computational Neuroscience Unit, University College London* (2003).

[3] Bengio, Y., Ducharme, R., and Vincent, P. A neural probabilistic language model. *In Proceedings of Neural Information Systems* (2000).

[4] Bengio, Y., Schwenk, H., Senécal, J.-S., Morin, F., and Gauvain, J.-L. Neural probabilistic language models. *Innovations in Machine Learning: Theory and Applications Edited by D. Holmes and L. C. Jain* (2005).

[5] Blei, D., Ng, A. Y., and Jordan, M. I. Latent dirichlet allocation. *Journal of Machine Learning Research 3* (2003), 9931022.

[6] Blei, D. M. Introduction to probabilistic topic models. *Communications of the ACM* (2011).

[7] Bordes, A., Glorot, X., Weston, J., and Bengio, Y. Towards open-text semantic parsing via multi-task learning of structured embeddings. *CoRR abs/1107.3663* (2011).

[8] Bresnan, J. Lexical functional syntax. *Blackwell* (2001).

[9] Chen, S., and Goodman, J. An empirical study of smoothing techniques for language modeling. *Technical report TR-10-98 Harvard University* (1998).

[10] Collobert, R. Deep learning for efficient discriminative parsing. *In Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)* (2011).

[11] Collobert, R., and Weston, J. A unified architecture for natural language processing: Deep neural networks with multitask learning. *In International Conference on Machine Learning (ICML 2008)* (2008).

[12] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. Natural language processing (almost) from scratch. *Journal of Machine Learning Research 12* (2011), 2493–2537.

[13] DeNero, J., Chiang, D., and Knight, K. Fast consensus decoding over translation forests. *In proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP* (2009), 567–575.

[14] Deschacht, K., Belder, J. D., and Moens, M.-F. The latent words language model. *Computer Speech and Language 26* (2012), 384–409.

[15] Du, J., He, Y., Penkale, S., and Way, A. MaTrEx: the DCU MT System for WMT 2009. *In Proceedings of the Third EACL Workshop on Statistical Machine Translation* (2009), 95–99.

[16] Du, J., and Way, A. An incremental three-pass system combination framework by combining multiple hypothesis alignment methods. *International Journal of Asian Language Processing 20*, 1 (2010), 1–15.

[17] Du, J., and Way, A. Using terp to augment the system combination for smt. *In Proceedings of the Ninth Conference of the Association for Machine Translation (AMTA2010)* (2010).

[18] Federico, M., Bertoldi, N., and Cettolo, M. Irstlm: an open source toolkit for handling large scale language models. *Proceedings of Interspeech* (2008).

[19] Gael, J. V., Vlachos, A., and Ghahramani, Z. The infinite hmm for unsupervised pos tagging. *The 2009 Conference on Empirical Methods on Natural Language Processing (EMNLP 2009)* (2009).

[20] Gasthaus, J., Wood, F., and Teh, Y. W. Lossless compression based on the sequence memoizer. *DCC 2010* (2010).

[21] Ghahramani, Z. An introduction to hidden markov models and bayesian networks. *International Journal of Pattern Recognition and Artificial Intelligence 15*, 1 (2001), 9–42.

[22] Ghahramani, Z., Jordan, M. I., and Smyth, P. Factorial hidden markov models. *Machine Learning* (1997).

[23] Goldwater, S., Griffiths, T. L., and Johnson, M. Contextual dependencies in unsupervised word segmentation. *In Proceedings of Conference on Computational Linguistics / Association for Computational Linguistics (COLING-ACL06)* (2006), 673–680.

[24] Good, I. J. The population frequencies of species and the estimation of population paramters. *Biometrika 40*, (3-4) (1953), 237–264.

[25] Hinton, G. E., McClelland, J. L., and Rumelhart, D. Distributed representations. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition(Edited by D.E. Rumelhart and J.L. McClelland) MIT Press 1* (1986).

[26] Kneser, R., and Ney, H. Improved backing-off for n-gram language modeling. *In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing* (1995), 181–184.

[27] Koller, D., and Friedman, N. Probabilistic graphical models: Principles and techniques. *MIT Press* (2009).

[28] Kumar, S., and Byrne, W. Minimum Bayes-Risk word alignment of bilingual texts. *In Proceedings of the Empirical Methods in Natural Language Processing (EMNLP 2002)* (2002), 140–147.

[29] Matusov, E., Ueffing, N., and Ney, H. Computing consensus translation from multiple machine translation systems using enhanced hypotheses alignment. *In Proceedings of the 11st Conference of the European Chapter of the Association for Computational Linguistics (EACL)* (2006), 33–40.

[30] Miller, G. A. Wordnet: A lexical database for english. *Communications of the ACM 38*, 11 (1995), 39–41.

[31] Mnih, A., and Teh, Y. W. A fast and simple algorithm for training neural probabilistic language models. In *Proceedings of the International Conference on Machine Learning* (2012).

[32] Mochihashi, D., Yamada, T., and Ueda, N. Bayesian unsupervised word segmentation with nested pitman-yor language modeling. *In Proceedings of Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP 2009)* (2009), 100–108.

[33] Murphy, K. P. Machine learning: A probabilistic perspective. *The MIT Press* (2012).

[34] Och, F., and Ney, H. A systematic comparison of various statistical alignment models. *Computational Linguistics 29*, 1 (2003), 19–51.

[35] Okita, T., and van Genabith, J. Minimum bayes risk decoding with enlarged hypothesis space in system combination. *In Proceedings of the 13th International Conference on Intelligent Text Processing and Computational Linguistics (CICLING 2012). LNCS 7182 Part II. A. Gelbukh (Ed.)* (2012), 40–51.

[36] Okita, T., and Way, A. Hierarchical pitman-yor language model in machine translation. *In Proceedings of the International Conference on Asian Language Processing (IALP 2010)* (2010).

[37] Okita, T., and Way, A. Pitman-Yor process-based language model for Machine Translation. *International Journal on Asian Language Processing 21*, 2 (2010), 57–70.

[38] Okita, T., and Way, A. Given bilingual terminology in statistical machine translation: Mwe-sensitve word alignment and hierarchical pitman-yor process-based translation model smoothing. *In Proceedings of the 24th International Florida Artificial Intelligence Research Society Conference (FLAIRS-24)* (2011), 269–274.

[39] Owczarzak, K., van Genabith, J., and Way, A. Evaluating machine translation with LFG dependencies. *Machine Translation 21*, 2 (2007), 95–119.

[40] Rabiner, L. R. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE 77*, 2 (1989), 257–286.

[41] Schwenk, H. Continuous space language models. *Computer Speech and Language 21* (2007), 492–518.

[42] Schwenk, H. Continuous space language models for statistical machine translation. *The Prague Bulletin of Mathematical Linguistics 83* (2010), 137–146.

[43] Schwenk, H., Rousseau, A., and Attik, M. Large, pruned or continuous space language models on a gpu for statistical machine translation. *In Proceeding of the NAACL workshop on the Future of Language Modeling* (2012).

[44] Sontag, D. Approximate inference in graphical models using LP relaxations. *Massachusetts Institute of Technology (Ph.D. thesis)* (2010).

[45] Sontag, D., and Roy, D. M. The complexity of inference in latent dirichlet allocation. *In Advances in Neural Information Processing Systems 24 (NIPS)* (2011).

[46] Steyvers, M., and Griffiths, T. Probabilistic topic models. *Handbook of Latent Semantic Analysis. Psychology Press* (2007).

[47] Stolcke, A. SRILM – An extensible language modeling toolkit. *In Proceedings of the International Conference on Spoken Language Processing* (2002), 901–904.

[48] Teh, Y. W. A hierarchical bayesian language model based on pitman-yor processes. *In Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics (ACL-06), Prague, Czech Republic* (2006), 985–992.

[49] Tromble, R., Kumar, S., Och, F., and Macherey, W. Lattice minimum bayes-risk decoding for statistical machine translation. *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing* (2008), 620–629.

[50] Veale, T. Exploding the creativity myth: The computational foundations of linguistic creativity. *London: Bloomsbury Academic* (2012).

[51] Wood, F., Archambeau, C., Gasthaus, J., James, L., and Teh, Y. W. A stochastic memoizer for sequence data. *In Proceedings of the 26th International Conference on Machine Learning* (2009), 1129–1136.